



Технології програмування C/Embedded

(Сертифікатна програма Інженерія вбудованих систем та інтернет речей
за участі компанії GlobalLogic)

Робоча програма навчальної дисципліни (Силабус)

1. Реквізити навчальної дисципліни

Рівень вищої освіти	Перший (бакалаврський)
Галузь знань	F (12) Інформаційні технології
Спеціальність	F2 (121) Інженерія програмного забезпечення; F7 (123) Комп'ютерна інженерія, F6 (126) Інформаційні системи та технології
Освітня програма	Комп'ютерні системи та мережі Інженерія програмного забезпечення комп'ютерних систем Інтегровані інформаційні системи Інформаційні управляючі системи та технології Інформаційне забезпечення робототехнічних систем Інженерія програмного забезпечення інформаційних систем Інженерія програмного забезпечення комп'ютерних систем
Статус дисципліни	Вибіркова
Форма навчання	очна (денна)
Рік підготовки, семестр	3 курс , осінній
Обсяг дисципліни	4 кредитів/120 год., лекцій 36 годин, лаб. роб. 18 год., СРС 66 год.
Семестровий контроль/ контрольні заходи	Залік, лабораторні роботи, МКР
Розклад занять	Згідно розкладу на осінній семестр поточного навчального року за адресою https://schedule.kpi.ua
Мова викладання	Українська
Інформація про керівника курсу / викладачів	Лектор: <u>Кондратенко Денис Юрійович, асистент кафедри ОТ, SW Engineer Global Logic</u> Лабораторні: асистент кафедри ОТ, Каплунов Артем Володимирович, kaplunov.artem@ill.kpi.ua
Розміщення курсу	Платформа дистанційного навчання «Сікорський» в середовищі Google Workspace for Education: Технології програмування на C/Embedded. https://classroom.google.com/c/NTI1OTI0NTq5Mzg5?cjc=je92wfd

2. Програма навчальної дисципліни

1. Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

Мета дисципліни «Технології програмування C/Embedded» – вивчення теоретичних основ, функціональних можливостей та принципів взаємодії основних компонентів вбудованих систем. Предметом вивчення кредитного модуля «Технології програмування C/Embedded» є архітектура та функціональні можливості сучасних вбудованих систем; системи введення/виведення; засоби та механізми пересилання та зберігання даних; знайомство з мовою низького рівня C та особливостями програмування для вбудованих систем.

Вивчення дисципліни підсилює наступні загальні та фахові компетенції:

Для спеціальності F2 (121) Інженерія програмного забезпечення

- Здатність до абстрактного мислення, аналізу та синтезу
- Здатність застосовувати знання у практичних ситуаціях
- Здатність вчитися і оволодівати сучасними знаннями
- Здатність працювати в команді
- Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення
- Здатність формулювати та забезпечувати вимоги щодо якості програмного забезпечення у відповідності з вимогами замовника, технічним завданням та стандартами
- Здатність дотримуватися специфікацій, стандартів, правил і рекомендацій в професійній галузі при реалізації процесів життєвого циклу
- Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя
- Здатність до алгоритмічного та логічного мислення
- Здатність розробляти та використовувати мережні технології
- Здатність розробляти та використовувати програмне забезпечення для високопродуктивних комп'ютерних систем
- Здатність використовувати принципи побудови, функціонування та узагальнену структуру мікропроцесорних систем та особливості програмного забезпечення мікропроцесорних систем та мікроконтролерів

Для спеціальності F7 (123) «Комп'ютерна інженерія»

- Здатність вчитися і оволодівати сучасними знаннями
- Здатність працювати в команді
- Здатність застосовувати законодавчу та нормативно правову базу, а також державні та міжнародні вимоги, практики і стандарти з метою здійснення професійної діяльності в галузі комп'ютерної інженерії.
- Здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення.
- Здатність використовувати засоби і системи автоматизації проектування до розроблення компонентів комп'ютерних систем та мереж, Інтернет додатків, кіберфізичних систем тощо.
- Здатність проектувати системи та їхні компоненти з урахуванням усіх аспектів їх життєвого циклу та поставленої задачі, включаючи створення, налаштування, експлуатацію, технічне обслуговування та утилізацію.
- Здатність проектувати, впроваджувати та обслуговувати високопродуктивні паралельні та розподілені комп'ютерні системи та їх складові з використання ПЛІС модулів і систем автоматизованого проектування
- Здатність розробляти, адаптувати, використовувати програмно-технічні засоби для збирання, оброблення та інтелектуального аналізу даних в комп'ютерних системах: паралельних,

розподілених, хмарних, безпечних, інтелектуальних, розумних, Інтернет речей, зокрема, з використанням штучного інтелекту.

Для спеціальності F6 (126) Інформаційні системи та технології

- Здатність до абстрактного мислення, аналізу та синтезу
 - Здатність застосовувати знання у практичних ситуаціях
 - Здатність розробляти та управляти проектами
 - Здатність аналізувати об'єкт проектування або функціонування та його предметну область
 - Здатність застосовувати стандарти в області інформаційних систем та технологій при розробці функціональних профілів, побудові та інтеграції систем, продуктів, сервісів і елементів інфраструктури організації
 - Здатність до проектування, розробки, налагодження та вдосконалення системного, комунікаційного та програмно апаратного забезпечення інформаційних систем та технологій, Інтернету речей (IoT), комп'ютерно інтегрованих систем та системної мережної структури, управління ними
 - Здатність управляти та користуватися сучасними інформаційно-комунікаційними системами та технологіями (у тому числі такими, що базуються на використанні Інтернет).
 - Здатність розробляти структури робототехнічних систем та проектувати функціональні пристрої роботів та робототехнологічних комплексів
 - Здатність використовувати знання основ електроніки і мікросхемотехніки, сучасної елементної бази при проектуванні мікропроцесорних пристроїв робототехнічних систем
- Здатність формувати нові конкурентоспроможні ідеї й реалізовувати їх у проектах (стартапах)

У відповідності до вищезазначеного, підсилені загальні та фахові компетенції дадуть наступні результати навчання:

- знання фундаментальних концепцій, парадигм і основних принципів функціонування вбудованих систем;
- здатність до використання теоретичних, логічних та арифметичних основ для розробки драйверів та програмного забезпечення для вбудованих систем і вміння їх застосовувати при вирішенні професійних завдань;
- здатність розробляти драйвери для окремих компонентів вбудованих систем, в тому числі, з використанням сучасних систем автоматизації проектування;
- здатність розробки та використання драйверів елементів архітектурозалежного рівня на основі знання загальних принципів організації та функціонування вбудованих систем;
- здатність брати участь у командній роботі по проектуванню драйверів та програмного забезпечення для вбудованих систем;
- здатність формувати та забезпечувати вимоги щодо надійності вбудованих систем у відповідності з вимогами замовника, технічним завданням та стандартами.

2. Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)

Для успішного засвоєння дисципліни «Технології програмування C/Embedded» студентам доцільно мати базові теоретичні та практичні знання, необхідні для проектування, складання та налагодження апаратно-програмних електронних систем, які набуті під час вивчення базових та професійно орієнтованих дисциплін та охоплюють такі напрями:

- основи операційних систем та інструментів командного середовища
- принципи комп'ютерної логіки та цифрової електроніки
- базові знання з архітектури комп'ютерів та їх складових
- фундаментальні навички алгоритмізації та програмування
- практичний досвід використання мов програмування та роботи з апаратними ресурсами

Знання та навички, отримані під час вивчення дисципліни «Технології програмування C/Embedded», можуть бути використані при опануванні подальших навчальних курсів та у практичній діяльності, зокрема в напрямках:

- програмування та налагодження вбудованих систем
- розробка застосувань для мікропроцесорних і мікроконтролерних платформ
- моделювання та програмування обчислювальних пристроїв на ПЛІС (FPGA)
- тестування та контроль якості програмно-апаратних комплексів
- проєктування інтелектуальних систем та IoT-рішень
- системне програмування та управління інфраструктурними ІТ-проєктами

3. Зміст навчальної дисципліни

Ленція 1. Вступ. С як мова програмування для вбудованих систем

Розділ 1. Препроцесор С

Лекція 2

Тема 1.1. Включення файлів

Тема 1.2. Заміна макросів

Тема 1.3. Обумовлене включення файлів

Розділ 2. Оператори

Лекція 2-3

Тема 2.1. Арифметичні операції

Тема 2.2. Реляційні та логічні оператори

Тема 2.3. Перетворення типів

Тема 2.4. Оператори приросту та спаду

Тема 2.5. Розрядні оператори

Тема 2.6. Оператори та вирази присвоєння

Тема 2.7. Вирази умов

Тема 2.8. Пріоритет і послідовність обчислення

Розділ 3. Структури

Лекція 4

Тема 3.1. Основні поняття про структури

Тема 3.2. Структури та функції

Тема 3.3. Масиви структур

Тема 3.4. Typedef

Розділ 4. Символьні константи

Лекція 5

Тема 4.1. Рядки в мові С

Тема 4.2. Оголошення рядків у С

Тема 4.3. Робота з рядками в С

Тема 4.4. Масиви рядків у С

Тема 4.5. Перетворення рядків

Тема 4.6. Обробка рядків

Тема 4.7. Символьні константи

Тема 4.8. Рядкові літерали

Розділ 5. Показчики

Лекція 6

Тема 5.1. Показчики й адреси

Тема 5.2. Показчики й аргументи функцій

Тема 5.3. Показчики та масиви

Тема 5.4. Арифметика адрес

Тема 5.5. Показчики на символи та функції

Тема 5.6. Масив показчиків; показчики на показчики

Розділ 6. Embedded та 'Hello World'

Лекція 7

Тема 6.1. Особливості програмування мікроконтролерів.

Лекція 8

Тема 6.2. Огляд та особливості архітектури STM32 та навчально-налагоджувальної плати

Лекція 9

Тема 6.3. Проблеми з IDE.

Тема 6.4. Налаштування компілятора STM32CubeIDE.

Лекція 10

Тема 6.5. Процес створення embedded-проекту.

Розділ 7. Аналіз коду Embedded C

Лекція 11

Тема 7.1. Пам'ять програм мікроконтролера.

Лекція 12

Тема 7.2. Аналіз ELF файлу за допомогою інструментів GNU.

Тема 7.3. Дизасемблер.

Розділ 8. Практична частина програмування для мікроконтролеру STM32

Лекція 13

Тема 8.1. Написання коду для ввімкнення та вимкнення світлодіода.

Лекція 14-15

Тема 8.2. Увімкнення периферійних пристроїв.

Тема 8.3. Обчислення адрес периферійних регістрів.

Розділ 9. Практичне закріплення (інтеграція знань)

Лекція 17

Тема 9.1. Повний приклад embedded-проекту STM32.

Тема 9.2. Типові помилки в Embedded C.

Розділ 10. Заключна частина

Лекція 18

Тема 10.1. Підсумок курсу Embedded C.

Тема 10.2. Подальші кроки в Embedded-розробці.

Тема 10.3. Рекомендації для подальшого навчання.

4. Навчальні ресурси та матеріали

4.1. Базова література

1. Технології програмування C/Embedded [Електронний ресурс]: Навчальний посібник для здобувачів ступеня бакалавра за спеціальністю 123 «Комп'ютерна інженерія» / уклад.: А. В. Каплунов, І. А. Клименко, Д. Ю. Кондратенко. – Київ : КПІ ім. Ігоря Сікорського, 2026. – 120 с. (Затверджено засіданням кафедри протокол № 10 від 15.02.2026 р.).
<https://comsys.kpi.ua/metodichni-vkazannya-po-disciplinam>.

2. Тарасенко-Клятченко, О. В. Програмування мовою C. Задачі до практичних занять з кредитного модуля «Програмування-1. Основи програмування» [Електронний ресурс] : навчальний посібник для студентів спеціальності «123 Комп'ютерна інженерія» / О. В. Тарасенко-Клятченко, Я. М. Клятченко, О. С. Михайлюк ; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 203 Кбайт). – Київ : КПІ ім. Ігоря Сікорського, 2021. – 43 с. – Назва з екрана.
<https://ela.kpi.ua/handle/123456789/45708>

3. Браян В. Керніган, Деніс М. Річі Мова програмування C, друге видання (Brian Kernighan and Dennis Ritchie The C Programming Language). Переклад: Віталій Цибуля [Електронний ресурс] :. – 232 с. – Режим доступу: <http://programming.in.ua/programming/c-language/227-book-programming-c-kernighan.html>.

4.2. Додаткова література

4. STMicroelectronics. Datasheet відповідного мікроконтролера STM32. Офіційна технічна документація виробника.

<https://www.st.com/en/microcontrollers-microprocessors/stm32-32-bit-arm-cortex-mcus.html>

5. STMicroelectronics. Reference Manual відповідного мікроконтролера STM32. Офіційний опис архітектури, периферійних модулів та регістрів мікроконтролера.

<https://www.st.com/en/microcontrollers-microprocessors/stm32-32-bit-arm-cortex-mcus.html>

6. STMicroelectronics. *Application Notes (ANxxxx)* з розроблення вбудованих систем на базі STM32. Методичні рекомендації та приклади практичного використання периферії мікроконтролерів STM32. <https://www.st.com/en/application-notes.html>

7. Brian W. Kernighan, Dennis M. Ritchie (1988) The C Programming Language, 2nd ed. Pearson.

8. Mark Siegesmund (2014) Embedded C Programming: Techniques and Applications of C and PIC MCUS. Newnes.

9. K. N. King (2008) C Programming: A Modern Approach, 2nd ed. W. W. Norton & Company

10. Manuel Bermudez (1998) Study Guide for C Programming: A Modern Approach. W. W. Norton & Company

11. Randal Bryant and David O'Hallaron (1994) Computer Systems: A Programmer's Perspective (2nd Edition). Addison Wesley

12. Stephen Kochan (2013) Programming in C (3rd Edition). Pearson.

13. Richard Heathfield and Lawrence Kirby (2000) C Unleashed. Sams.

4.3. Інформаційні ресурси

1. Дистанційний курс на платформі дистанційного навчання «Сікорський» в середовищі Google Workspace for Education: Технології програмування на C/Embedded. <https://classroom.google.com/c/NTI1OTI0NTg5Mzg5?cjc=jey2wfd>. На ресурсі розміщено додаткові теоретичні матеріали та приклади виконання практичних задач, презентації та відеозаписи лекцій, перелік питань та завдання до самостійного опрацювання, матеріали для підготовки до заліку.

2. Fastbit Embedded Brain Academy – YouTube. –

<https://www.youtube.com/channel/UCa1REBV9hyrzGp2mjCagBg>

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компоненту)

Розподіл навчального часу за видами занять і завдань з дисципліни згідно з робочим навчальним планом. На кредитний модуль виділено 120 годин та 4 кредитів – 18 лекцій (36 годин), 9 практичних робіт (18 годин), самостійна робота (66 годин), модульна контрольна робота (МКР).

Лекційні заняття

Для досягнення мети навчальної дисципліни «Технології програмування C/Embedded» навчальний процес має бути зорієнтований на поєднання теоретичних знань і практичних навичок програмування вбудованих систем.

У лекційному матеріалі доцільно зосередитись на:

- основах мови програмування C у контексті вбудованих застосунків;
- принципах організації програмного забезпечення для мікроконтролерів;
- засобах взаємодії програмного коду з апаратними ресурсами;
- інструментах розробки та налагодження програм для вбудованих систем;
- практичних прикладах застосування вбудованих платформ у сучасних IT-проектах.

Лабораторні роботи

Практична частина дисципліни реалізується через лабораторні роботи, метою яких є набуття вмінь і навичок застосування принципів розробки програмного забезпечення для вбудованих

систем та їхніх функціональних модулів. Для цього використовуються сучасні середовища та інструменти програмування (Vim, Eclipse, Cube IDE), а також апаратні платформи (STM32 Discovery, STM StarterKit GlobalLogic).

Мета лабораторних робіт полягає у набутті студентами практичних навичок програмування вбудованих систем мовою **C** на базі мікроконтролерів STM32 з використанням навчально-налагоджувальної плати STM32 Discovery, а також формуванні розуміння принципів розроблення embedded-програмного забезпечення на рівні апаратури та програмного коду.

Ціллю виконання лабораторних робіт є формування ґрунтовного розуміння процесів розроблення програмного забезпечення для мікроконтролерів у середовищі *bare-metal*, набуття практичного досвіду роботи з архітектурою ARM Cortex-M, периферійними модулями мікроконтролера та інструментами розробки (STM32CubeIDE, компілятор GCC, засоби налагодження).

У процесі виконання лабораторних робіт студенти здобувають необхідні знання та навички для участі в проєктах з розроблення вбудованих систем на ролі *embedded-розробника*, а також базові уявлення про подальший розвиток у напрямках *firmware development*, *low-level programming* та *embedded-DevOps*.

Тематика лабораторних робіт:

Лабораторна робота 1. Препроцесор C.

Лабораторна робота 2. Оператори.

Лабораторна робота 3. Структури.

Лабораторна робота 4. Символьні константи.

Лабораторна робота 5. Показчики.

Лабораторна робота 6. STM32F4-DISCO. Розгортання середовища та ПЗ.

Лабораторна робота 7. Прошивка GL Starter Kit. STM32CubeIDE & Qemu.

Лабораторна робота 8. Маніпуляція світлодіодами за допомогою програмованої кнопки.

6. Самостійна робота здобувача вищої освіти денної форми навчання

Види самостійної роботи (66 годин):

- підготовка до аудиторних занять, виконання поточних домашніх завдань та опрацювання матеріалів лекцій (**0,5 годин x 36 годин = 18 годин**);
- виконання індивідуального завдання до лабораторних робіт, розв'язок задач, підготовка та оброблення проведення розрахунків за первинними даними, отриманими на лабораторних заняттях, оформлення протоколу, оформлення звіту до практичної роботи (1 година x 18 годин практичних занять (8 лабораторних робіт) = **18 годин**);
- підготовка до МКР (**2 години**);
- підготовка до заліку (**6 годин**);

самостійне опрацювання теоретичного матеріалу, розгортання програмного середовища та завантаження вихідних кодів програмного забезпечення для виконання практичних робіт, робота з системою контролю версій Git, підготовка презентації (**22 години**).

Теми на самостійне опрацювання

Розділ 2. Оператори

Тема 2.5. Розрядні оператори

Тема 2.6. Оператори та вирази присвоєння

Тема 2.7. Вирази умов

Тема 2.8. Пріоритет і послідовність обчислення

Розділ 4. Символьні константи

Тема 4.5. Перетворення рядків

Тема 4.6. Обробка рядків

Тема 4.7. Символьні константи

Тема 4.8. Рядкові літерали

Розділ 5. Показчики

Тема 5.4. Арифметика адрес

Тема 5.5. Показчики на символи та функції

Тема 5.6. Масив показчиків; показчики на показчики

Питання на самостійне опрацювання для практичної частини курсу:

Розділ 9. Практичне закріплення (інтеграція знань)

Тема 9.1. Повний приклад embedded-проєкту

1. Проаналізувати структуру простого embedded-проєкту для мікроконтролерів STM32. Визначити призначення основних файлів проєкту (main.c, startup-файл, linker script).

2. На основі документації STM32 пояснити роль:

- startup-коду;
 - таблиці векторів переривань;
 - linker script у процесі формування прошивки.
3. Використовуючи Reference Manual обраного STM32, описати:
- етапи ініціалізації тактування мікроконтролера;
 - наслідки некоректного налаштування системного тактування.

4. На прикладі GPIO:

- знайти в документації регістри керування портами введення/виведення;
- пояснити послідовність увімкнення тактування периферії;
- описати процес налаштування виводу на вихід.

5. Реалізувати або проаналізувати код керування світлодіодом у нескінченному циклі та пояснити принцип його роботи на рівні регістрів.

Тема 9.2. Типові помилки початківців в Embedded C

6. Навести приклади помилок, пов'язаних із некоректним обчисленням або використанням адрес периферійних регістрів.

7. Пояснити призначення ключового слова `volatile` та описати наслідки його відсутності при роботі з апаратними регістрами.

8. Проаналізувати використання нескінченних циклів (`while(1)`) в embedded-програмах:

- у яких випадках вони є необхідними;
- які потенційні проблеми можуть виникати.

9. Дослідити підхід до реалізації затримок типу *busy wait*:

- переваги;
- недоліки;
- альтернативні рішення.

10. На прикладах з документації або літератури пояснити, як оптимізація компілятора може впливати на поведінку embedded-коду та призводити до «неочікуваних» результатів.

Розділ 10. Заключна частина

Тема 10.1. Підсумок курсу Embedded C

11. Узагальнити поняття embedded-систем та навести приклади їх застосування в сучасній техніці.

12. Пояснити архітектурні особливості мікроконтролерів у порівнянні з мікропроцесорами.

13. Описати принципи bare-metal програмування та його відмінності від програмування з використанням операційної системи.

14. Проаналізувати роль мови програмування C в embedded-розробці та причини її широкого використання.

15. Сформулювати перелік практичних навичок, якими володіє студент після завершення курсу.

Тема 10.2. Подальші кроки в Embedded-розробці

16. На основі документації STM32 дослідити принцип роботи апаратних переривань та їх призначення.

17. Описати можливості таймерів мікроконтролера та навести приклади їх практичного використання.

18. Порівняти інтерфейси UART, SPI та I2C:

- призначення;
- переваги та обмеження кожного з них.

19. Проаналізувати підходи HAL та Register Level programming, визначити їх сильні та слабкі сторони.

20. Ознайомитися з концепцією RTOS (на прикладі FreeRTOS) та визначити, у яких задачах її використання є доцільним.

21. Дослідити основні методи налагодження embedded-програм:

- breakpoints;
- перегляд регістрів;
- аналіз пам'яті.

Тема 10.3. Рекомендації для самостійного навчання

22. Ознайомитися зі структурою технічної документації STM32 (Datasheet, Reference Manual, Application Notes) та пояснити призначення кожного документа.

23. На прикладі однієї Application Note описати, як вона допомагає у вирішенні практичних embedded-задач.

24. Проаналізувати рекомендовані книги або навчальні ресурси з embedded-програмування та обґрунтувати їхню практичну цінність.

25. Обґрунтувати роль практики як ключового елементу професійного розвитку embedded-розробника.

Матеріали для самостійного опрацювання, питання для самоконтролю та контрольні запитання знаходяться на платформі дистанційного навчання «Сікорський» (Classroom) <https://classroom.google.com/c/NzI3OTc5MzE2MDk4?cjc=qag146m>.

3. Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

Дедлайни. Для виконання лабораторних робіт встановлюються дедлайни. Дедлайни уточнюються і оприлюднюються викладачем на початку семестру.

Заохочувальні бали (бонуси) виставляються за: активну участь на лекціях та семінарах, наприклад участь в дискусіях; виконання домашніх завдань, ведення конспекту, підготовка повідомлення з презентацією по одній із тем СРС дисципліни тощо. Кількість заохочуваних балів не більше 10.

Заохочувальні бали також можуть нараховуватися за прояви **soft skills** під час виконання контрольних заходів дисципліни (лабораторних робіт, поточних контрольних робіт, модульних контрольних робіт). До таких проявів належать, зокрема, старанність під час виконання завдань, дотримання встановлених термінів подання робіт, відповідальне ставлення до навчального процесу, самостійність та якість оформлення результатів. Бали нараховуються в розмірі **до 1 балу за одне завдання** за рішенням викладача.

Лабораторні роботи. Виконанню кожної лабораторної роботи передуює виконання індивідуального завдання і оформлення його у вигляді протоколу. Протокол виконання лабораторного завдання має бути завантажено в Classroom не пізніше, ніж за добу до лабораторного заняття. До протоколу додаються посилання на Git з кодом програм (за необхідності) і посилання на всі необхідні супровідні матеріали для захисту, зокрема на відеозапис виконання лабораторного завдання (за необхідності - в дистанційному режимі, за використання віддаленого устаткування, тощо). Студент, який прийшов на заняття без підготовленого протоколу до захисту лабораторної роботи не допускається. Першим етапом студент захищає результати виконання індивідуального завдання до лабораторної роботи, на другому етапі – захищає результати виконання завдання шляхом усного опитування або тестування. У дистанційному

режимі більшість лабораторних робіт супроводжуються тестами для оцінки вивченого теоретичного та практичного матеріалу замість усного опитування. Викладач залишає за собою право провести усне опитування у разі, якщо результати виконання лабораторної роботи є сумнівними або потребують уточнення. Бали отримані за виконання лабораторної роботи, оформлення протоколу та захист результатів складають оцінку за лабораторну роботу. Тестування проводиться на лабораторному занятті після перевірки результатів виконання індивідуальної роботи. Студент, який не виконав індивідуальне завдання до лабораторної роботи до тесту (захисту результатів) не допускається.

Бали, які студент може отримати за виконання кожної лабораторної роботи наведено в таблиці 1 оцінювання семестрових робіт, розділ 8 силабуса.

Експрес-тести. Окремі теми лекційних занять супроводжуються короткими експрес-тестами (на 15 хвилин), які включають матеріал вивченої теми та питання, які задані на самостійне вивчення. Бали отримані за тест входять в семестрову рейтингову оцінку. Поточні тести не перескладаються і не є обов'язковими для допуску до семестрового контролю.

Модульна контрольна робота. Модульна контрольна робота пишеться на лекційному занятті без застосування допоміжних засобів (мобільні телефони, планшети та ін.); результат пересилається до відповідної директорії Google -диску через Google форму. Модульна контрольна не виконується поза визначений термін і не перепишується. Бали за залікову роботу додаються до балів за лабораторні роботи та модульну контрольну роботи і складають семестрову рейтингову оцінку.

Семестровий контроль. Загальною умовою допуску до семестрового контролю (заліку) є зарахування всіх лабораторних робіт, МКР та рейтинг не менше 60 балів. Таким чином мінімальна оцінка, яку повинен отримати студент для допуску до семестрового контролю (заліку) дорівнює 60 балів за виконання всіх поточних робіт за семестр.

Рейтингова (семестрова) оцінка визначається як сума балів за модульну контрольну роботу, балів за індивідуальну семестрову роботу (лабораторні роботи) та експрес тести. Кількість балів за індивідуальну роботу за семестр дорівнює від 38 до 60 балів. Максимальна оцінка за МКР дорівнює 30 балів.

Відповідно до Положення про систему оцінювання результатів навчання в КПІ ім. Ігоря Сікорського https://osvita.kpi.ua/sites/default/files/downloads/Pologennia_RSO_2022.pdf пп. 3.7 – 3.11 передбачено наступне:

Здобувачі, які виконали всі умови допуску до заліку та мають рейтингову оцінку менше 60 балів, а також здобувачі, які бажають підвищити свою рейтингову оцінку, на останньому за розкладом занятті складають залік у вигляді виконання залікового завдання або співбесіди (за вибором викладача).

У разі виконання залікової контрольної роботи або співбесіди, рейтингова (семестрова) оцінка, визначається, як **сума балів за індивідуальну семестрову роботу (що відповідають балам за лабораторні роботи протягом семестру, див. табл. 1) та оцінка за залікову контрольну роботу (співбесіду).** Всі інші бали, які набрані протягом семестру (окрім балів за лабораторні роботи) анулюються.

Правила відвідування занять. Відвідування занять є вільним, бали за присутність на лекціях та лабораторних заняттях не додаються, відповідно штрафні бали не передбачаються. Втім, вагома частина рейтингу формується через активну участь у заходах на лабораторних заняттях.

Дистанційний режим навчання. У разі запровадження обмежень на відвідування університету, пов'язаних з введенням карантину або режиму воєнного стану в державі, освітній процес здійснюється у дистанційному режимі відповідно до Положення про дистанційне навчання в КПІ ім. Ігоря Сікорського (<https://osvita.kpi.ua/index.php/node/188>), Регламенту організації освітнього процесу в дистанційному режимі (<https://profkom.kpi.ua/reglament-organizatsiyi-osvitnogo-protsesu-v-distantsiynomu-rezhimi>) та Регламенту проведення семестрового контролю в дистанційному режимі (<https://osvita.kpi.ua/node/148>). У режимі дистанційного навчання заняття відбуваються у вигляді онлайн-конференції на платформах BigBlueButton, Google Meet, Zoom. Посилання на конференцію видається на початку семестру і розміщується в АС «Електронний кампус». З метою забезпечення якісної підготовки здобувачів, дистанційний курс дисципліни розміщено на

Платформа дистанційного навчання «Сікорський» (<https://www.sikorsky-distance.org>). Результати оцінювання висвітлюють у АС «Електронний кампус» на особистій сторінці здобувача (<https://ecampus.kpi.ua>).

Правила поведінки на заняттях. На заняттях слід дотримуватись норм етичної поведінки визначених у Кодексі честі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» (<https://kpi.ua/code>). На території університету здобувачі мають поводити себе відповідно до Правил внутрішнього розпорядку (<https://kpi.ua/admin-rule>).

Визнання результатів навчання, набутих у неформальній/інформальній освіті. Положення про визнання результатів навчання, набутих у неформальній / інформальній освіті (<https://osvita.kpi.ua/index.php/node/179>) регламентує визнання результатів навчання, набутих у неформальній/інформальній освіті:

- у разі навчання на неформальній освіті, яку студент обрав самостійно і хоче зарахувати результати навчання в рамках дисципліни, проходиться процедура валідації, що передбачає подання здобувачем заяви на ім'я декана, декларації підтверджувальних документів; рішення про визнання чи не визнання приймається комісією у складі завідувача кафедри, викладача, гаранта освітньо-професійної програми.

- за рішенням викладача окремі змістовні модулі або теми дисципліни можуть бути зараховані на підставі результатів проходження здобувачем онлайн-курсу або іншого елементу неформальної освіти; перелік завдань та елементів неформальної освіти визначається викладачем заздалегідь і розміщується на платформі дистанційного навчання «Сікорський», про що студенти повідомляються на початку семестру; у цьому випадку здобувач звільняється від виконання відповідних завдань і отримує максимальний бал згідно з рейтинговою системою оцінювання.

Політика використання штучного інтелекту. Використання штучного інтелекту регламентується «Політикою використання штучного інтелекту для академічної діяльності в КПІ ім. Ігоря Сікорського» (<https://osvita.kpi.ua/node/1225>). Усі завдання, як під час виконання навчальних завдань з дисципліни, так і індивідуальні завдання, мають бути результатом власної оригінальної роботи здобувача. Використання ШІ для автоматичної генерації відповідей без подальшого їх аналізу та доопрацювання заборонено.

Порушення академічної доброчесності: використання плагіату. У разі виявлення фактів використання плагіату під час виконання навчальних завдань, студент втрачає право на подальше виконання поточних завдань у межах навчальної дисципліни до з'ясування обставин порушення.

Студент зобов'язаний аргументовано пояснити причини використання плагіату, а також засвідчити готовність дотримуватись принципів академічної доброчесності в подальшому навчанні (**письмово у вигляді пояснювальної записки**); повторно виконати завдання або, за потреби, виконати додаткове завдання відповідно до бачення викладача з урахуванням вимог щодо самостійності.

Зазначені дії регламентуються Кодексом честі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» (<https://kpi.ua/code>).

8. Види контролю та рейтингова система оцінювання результатів навчання (PCO)

Семестровий рейтинг студента з кредитного модуля розраховується, виходячи із 100-бальної шкали. Рейтинг складається з балів, що студент отримує за виконання 4 лабораторних робіт $R_{\text{ЛР}}$, модульної контрольної роботи $R_{\text{МКР}}$ та експерс-тестів $R_{\text{ЕТ}}$.

Максимальна кількість балів за лабораторні роботи складає 60 балів, тобто $R_{\text{ЛР}} = 60$.

Узагальнені критерії оцінювання лабораторних робіт наступні:

Критерії оцінювання лабораторних робіт наступні:

- *Протокол* (2 б): Відповідність протоколу лабораторній роботі та встановленим вимогам до оформлення, коректність розміщення протоколу в Classroom/Git, повнота та логічність викладення матеріалу:

0 балів – протокол відсутній або не відповідає лабораторній роботі та встановленим вимогам, зокрема містить скриншоти та інші ілюстративні матеріали, які не дозволяють однозначно ідентифікувати середовище та користувача виконання роботи.

1 бал – протокол частково відповідає лабораторній роботі та вимогам до оформлення; матеріал подано неповно або з порушенням логіки викладення, результати виконання неповні, відсутні висновки.

2 бали – протокол повністю відповідає лабораторній роботі та встановленим вимогам до оформлення; коректно розміщений у Classroom/Git; матеріал викладено повно й логічно, наявні змістовні висновки; надані ілюстративні матеріали (скриншоти, фрагменти консолі тощо) є **автентичними, належать автору роботи та містять ідентифікаційні ознаки користувача середовища виконання**.

– *Виконання завдання* (4 б): розроблені моделі працюють правильно у програмному або апаратному середовищі, відповідають вимогам завдання (2 бали), студент показує власний репозиторій з усіма матеріалами лабораторної роботи; репозиторій містить логічну структуру файлів та історію комітів, що демонструє послідовність роботи (1 бал), студент пояснює та аргументує отримані результати, демонструє розуміння виконаного завдання (1 бал);

– *Опитування/тестування* (4 б) за тематикою лабораторної роботи для зарахування теоретичної частини роботи, відповіді на додаткові теоретичні запитання викладача: 0 – 4 бали.

Усний захист лабораторних робіт за баченням викладача може бути замінений на тестування, зокрема у вигляді автоматизованого тестування на платформі Google Workspace for Education / Moodle. Оцінка за поточний тест – 4 бали.

Детальний підхід до оцінювання кожної лабораторної роботи наведений в таблиці 1.

Таблиця 1. Деталізація балів за поточні роботи за семестр

Назва заняття	Форма контролю	Кількість балів	Допуск до заліку автоматом	Всього балів
Лабораторна робота 1	Виконання завдання	4	5	8
	Опитування	2		
	Протокол	2		
Лабораторна робота 2	Виконання завдання	4	5	8
	Опитування	2		
	Протокол	2		
Лабораторна робота 3	Виконання завдання	4	5	8
	Опитування	2		
	Протокол	2		
Лабораторна робота 4	Виконання завдання	4	5	8
	Опитування	2		
	Протокол	2		
Лабораторна робота 5	Виконання завдання	4	5	8
	Опитування	2		
	Протокол	2		
Лабораторна робота 6	Виконання завдання	2	2	4

	Протокол	2		
Лабораторна робота 7	Виконання завдання	4	5	8
	Опитування	2		
	Протокол	2		
Лабораторна робота 8	Виконання завдання	4	5	8
	Опитування	2		
	Протокол	2		
Індивідуальна робота	-	-	-	60
Експрес-тести на лекціях	2 x 5	10	5	10
МКР	МКР1	15	9	30
Всього балів		100	60	100

Максимальна кількість балів за МКР дорівнює $R_3 = 30$ балів.

Оцінка за МКР знижується за:

- некоректне оформлення роботи на віддаленому репозиторії, помилки в файлах збірки;
- відсутність коментарів в програмному коді та оформленні алгоритмів;
- відсутність коментарів та пояснень під час розрахунків.

Календарна атестація студентів (на 8 та 14 тижнях семестрів) з дисципліни проводиться за значенням поточного рейтингу студента на час атестації. Якщо значення цього рейтингу не менше 50 % від максимально можливого на час атестації, студент вважається атестованим. В іншому випадку в атестаційній відомості виставляється «неатестовано».

Кількість балів, що отримує студент за семестр визначається за формулою:

$$R = R_{\text{ПР}} + R_{\text{ЕТ}} + R_{\text{МКР}}$$

Якщо студент пише залікову контрольну роботу (проходить співбесіду) для підвищення балів його рейтингова (семестрова) оцінка визначається за формулою:

$$R = R_{\text{ПР}} + R_3$$

Максимальна кількість балів за семестр не перевищує **$R = 100$** .

З урахуванням одержаної суми балів кінцева оцінка визначається за таблицею 2.

Таблиця 2. Визначення семестрової оцінки

Кількість балів	Оцінка
100-95	Відмінно
94-85	Дуже добре
84-75	Добре
74-65	Задовільно
64-60	Достатньо
Менше 60	Незадовільно
Не виконані умови допуску	Не допущено

Робочу програму навчальної дисципліни (силабус):

Складено, асистент, Каплунов Артем Володимирович

Увалено кафедрою ОТ (протокол № 18 від 24.06.2026)

Погоджено методичною комісією ФІОТ (протокол № 12 від 26.06.2026)